Problem (1) [10 points] Write (in C++) a midpoint() member function for the class below (i.e. inside the class). The midpoint can be found by averaging (i.e. mean) the start and end values. Simply write the definition of midpoint(), you do not need to declare it.

class Problem1 {
private:
    double xStart;
    double xEnd;
public:
    Problem1();
};
**Problem (2)** [10 points] Write (in C++) both a deconstructor and a non-default constructor for the following class. The non-default constructor should take in a single integer as an input and make an array of that length filled with 0’s.

```cpp
class Problem2 {
private:
    int* vector;
public:
    Problem2();
};
```

**Problem (3)** [10 points] Make a child class for Problem3 below, and override the sayHello() function. Then modify main to make it display this new sayHello() function you created while keeping x as a Problem3*. You cannot declare/use any variables other than x.

```cpp
class Problem3 {
public:
    void sayHello();
};

void Problem3::sayHello()
{
    cout << "Take a hike!" << endl;
}

int main()
{
    Problem3* x = new Problem3();
    x->sayHello();
    delete x;
}
```
Problem (4) [10 points] Write (in C++) a copyArray() function, which returns a dynamically created array identical to the (non-dynamic) array passed in as input. You may assume “SIZE” is a global variable of value 5.

    int original[SIZE] = {1, 5, 3, 6, 3}
    int* copy = copyArray(original, SIZE);

    for (int i = 0; i < SIZE; i++)
    {
        if (copy[i] != original[i])
        {
            cout << "You will lose points.\n";
        }
    }

Problem (5) [10 points] Write (in C++) an overload function for the “+” operator for the following “Money” class to add together the amounts. You should never have “cents” over 100. This time you do need to declare the function in the class (you can draw an arrow from some text into the class given below) along with the definition.

    class Money {
    private:
        int dollars;
        int cents;
    public:
        Money();
        double getValue(); // 10 dollars and 75 cents, this would return 10.75
    };
Problem (6) [10 points] Write (in C++) an average() function that takes as input an array of Money (as defined in problem 5) and an int denoting the size of the array. This function should correctly find and return the average amount of money across all values in the array passed in. You may assume “SIZE” is a global variable which has value 10. An example code of its use is given below:

```c++
Money showMeThe[SIZE];
// magically initialize array
double mean = average(showMeThe, SIZE);
```

Problem (7) [10 points] Write (in C++) the givePointer() function whose use is shown below. You should give as input to this function one int and simply give back a pointer to that int. Will you need to use delete on this? Explain why or why not.

```c++
int x = 7;
int* xPtr = givePointer(x);
```
Problem (8) [10 points] Find 3 possible places for errors in the following code. Assume this is all the code except for namespaces and includes. Explain specifically what causes each error and whether it is a syntax or logic error:

class Problem8 {
    private:
        double y;
};

class Problem8child {
    public:
        getY();
};

double Problem8child::getY()
{
    return y;
}

int main()
{
    Problem8child x;
    cout << getY();
    return 0;
}
Problem (9) [10 points] Write (in C++) a recursive function flip() that will cout all the possible coin heads/tails combinations of flipping a coin “n” times (the first argument is a string (hint... hint...) and the second input is the int n). [Hint: use “+=” operator with strings to add letters to an existing string]

```cpp
int main()
{
    flip("", 3);
    // above should cout (in any order, each on a separate line):
    // HHH, HHT, HTH, THH, TTH, THT, HTT, TTT
}
```

Problem (10) [10 points] Assume the file “numbers.txt” has only integers in it. Write a segment of code in C++ (i.e. pretend you are writing somewhere in main) that finds the average of all numbers in this file. You must check for errors when trying to open the file. You may assume all includes have been done for you.
Problem (11) [10 points] Write a short paragraph describing the differences between the functions foo() and bar() provided below. If the input variable “x” was a user-defined class (e.g. Point, Complex, Ship, Money, etc.) instead of an int, what constructors (if any) would be run for each function? (Ignore the fact that “x = 2” might make no sense for a user-defined class.)

void foo(int x)
{
    x=2;
}

void bar(int &x)
{
    x=2;
}

Problem (12) [10 points] Write a loop using the variable “i” that loops over the range of i values shown (you may use any additional variables if you want). An example is provided below:

(Example) int i: 0, 1, 2, 3, 4
Answer:
for(int i=0; i < 5; i++)
    or
int i=0;
int dummy = 1;
while(i < 5)
{
    i=i+dummy;
}

(a) int i: 1, 10
(b) int i: 5, 6, 7, 8, 9, 10
(c) int i: 2, 4, 8, 16, 32
(d) int i: 1234, 123, 12, 1
(e) int i: 1, 1, 2, 3, 5, 8, 13, 21